Appendix J

Dies ist ein Pre-Release des Windows 95-Styleguides.
Microsoft Corporation: The Windows Interface Guideline for Software Design. Redmond (WA) [Microsoft Press], 1995, ISBN 1-55615-679
Die deutsche Fassung Microsoft Corp.: Die Windows Oberfläche. Leitfaden zur Softwaregestaltung. Unterschleißheim [Microsoft Press], 1995, ISBN 3-86063-226-4 ist vergriffen

The Windows Interface Guidelines — A Guide for Designing Software

Microsoft® Windows®

February 1995

This is a preliminary release of the documentation. It may be changed substantially prior to final commercial release. This document is provided for informational purposes only and Microsoft Corporation makes no warranties, either expressed or implied, in this prerelease document.

Aktualisierungen siehe ggf.:
http://msdn.microsoft.com/UI/default.asp und http://msdn.microsoft.com/UI/winuidraft.asp [06.09.1999]

Uwe Haupt haupt@tzi.de
Universität Bremen
Technologie-Zentrum Informatik
Institut für Software-Ergonomie und Informationsmanagement 07.09.1999

There is no keyboard interface for direct manipulation transfers.

You can support direct manipulation transfers to any visible object. The object (for example, a window or icon) need not be currently active. For example, the user can drop an object in an inactive window. This action activates the window. If an inactive object cannot accept a direct manipulation transfer, it (or its container) should provide feedback to the user.

How the transferred object is integrated and displayed in the drop destination is determined by the destination's context. A dropped object can be incorporated either as native data, as an OLE object, or as a partial form of the object such as its properties or a transformed object. You determine whether to add to or replace an existing selection based on the context of the operation, using such factors as the formats available for the object, the destination's purpose, and any user-supplied information such as the specific location that the user drops or commands (or modes) that the user has selected. For example, an application can supply a particular type of tool for copying the properties of objects.

Default Drag and Drop

Default drag and drop transfers an object using mouse button 1. How the operation is interpreted is determined by what the destination defines as the appropriate default operation. As with the command method, the destination determines this based on information about the object (and the formats available for the object) and the context of the destination itself. Avoid defining a destructive operation as the default. When that is unavoidable, display a message box to confirm the intentions of the user.

Using this transfer technique, the user can directly transfer objects between documents defined by your application as well as to system resources, such as folders and printers. Support drag and drop following the same conventions the system supports: the user presses button 1 down on an object, moves the mouse while holding the button down, and then releases the button at the destination. For the pen, the destination is determined by the location where the user lifts the pen tip from the screen.

The most common default transfer operation is Move, but the destination (dropped on object) can reinterpret the operation to be whatever is most appropriate. Therefore, you can define a default drag and drop operation to be another general transfer operation such as Copy or Link, a destination specific command such as Print or Send To, or even a specialized form of transfer such as Copy Properties.

Nondefault Drag and Drop

Nondefault drag and drop transfers an object using mouse button 2. In this case, rather than executing a default operation, the destination displays a pop-up menu when the user releases the mouse button, as shown in Figure 5.13. The pop-up menu contains the appropriate transfer completion commands.

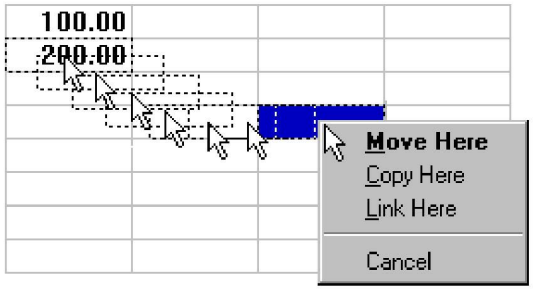

Figure 5.13 A nondefault drag and drop operation

The destination always determines which transfer completion commands to include on the resulting pop-up menu, usually factoring in information about the object supplied by the source location.

The form for nondefault drag and drop transfer completion verbs follows similar conventions as the Paste command. Use the common transfer completion verbs, Move Here, Copy Here, and Link Here, when the object being transferred is native data of the destination. When it is not, include the short type name. You can also display alternative completion verbs that communicate the context of the destination; for example, a printer displays a Print Here command. For commands that support only a partial aspect or a transformation of an object, use more descriptive indicators—for example, Copy Properties Here, or Transpose Here.

Use the following general form for nondefault drag and drop transfer commands.

[*Command Name*] [*object type* | *object name*] **Here** [**as** *object type*]

The following summarizes command forms for nondefault transfer completion commands.

Command	**Function**
Move Here	Moves the selected object to the destination as native content (data).
Copy Here	Creates a copy of the selected object in the destination as native content.
Link Here	Creates a data link between the selected object and the destination. The original object's value is integrated or transformed as native data within the destination, but remains linked to the original object so that changes to it are reflected in the destination.
Move [*short type name*] Here Copy [*short type name*] Here	Moves or copies the selected object as an OLE embedded object. The OLE embedded object is displayed in its content presentation and can be activated directly within the destination.
Link [*short type name*] Here	Creates an OLE linked object displayed as a picture of the selected object. The representation is linked to the selected object so that any changes to the original object will be reflected in the destination.
Move [*short type name*] Here as Icon Copy [*short type name*] Here as Icon	Moves or copies the selected object as an OLE embedded object and displays it as an icon.
Create Shortcut Here	Creates an OLE linked object to the selected object; displayed as a shortcut icon. The representation is linked to the selected object so that any changes to the original object will be reflected in the destination.

Table 11.1 Descriptive Text for Paste Special

Function	Descriptive text
Paste as an embedded object.	"Inserts the contents of the Clipboard into your document so you that you may activate it using *CompanyName ApplicationName*."
Paste as an embedded object so that it appears as an icon.	"Inserts the contents of the Clipboard into your document so you that you may activate it using *CompanyName ApplicationName application*. It will be displayed as an icon."
Paste as native data.	"Inserts the contents of the Clipboard into your document as *native type name [and optionally an additional Help sentence]*."
Paste as a linked object.	"Inserts a picture of the contents of the Clipboard into your document. Paste Link creates a link to the source file so that changes to the source file will be reflected in your document."
Paste as a linked object so that it appears as a shortcut icon.	"Inserts a Shortcut icon into your document which represents the contents of the Clipboard. A link is created to the source file so that changes to the source file will be reflected in your document."
Paste as linked native data.	"Inserts the contents of the Clipboard into your document as *native type name*. A link is created to the source file so that changes to the source file will be reflected in your document."

The Paste Link, Paste Shortcut, and Create Shortcut Commands

If linking is a common function in your application, you can optionally include a command that optimizes this process. Use Paste Link to support creating a linked object or linked native data. When using the command to create a linked object, include the name of the object preceded by the word "to"—for example, "Paste Link to Latest Sales." Omitting the name implies that the operation results in linked native data.

Use a Paste Shortcut command to support creation of a linked object that appears as a shortcut icon. You can also include a Create Shortcut command that creates a shortcut icon in the container. Apply these commands to containers where icons are commonly used.

Direct Manipulation

You can also support direct manipulation interaction techniques, such as drag and drop, for creating OLE embedded or linked objects. When the user drags a selection into a container, the container application can interpret the operation using information supplied by the source, such as the selection's type and format and by the destination container's own context, such as the container's type and its default transfer operation. For example, dragging a spreadsheet cell selection into a word-processing document can result in an OLE embedded table object. Dragging the same cell selection within the spreadsheet, however, would likely result in simply transferring the data in the cells.

Similarly, the destination container in which the user drops the selection also determines whether the dragged object creates an OLE linked object. For nondefault OLE drag and drop, the container application displays the appropriate transfer commands on the resulting pop-up menu and carries out the operation corresponding to the user's choice on that menu. The choices may include multiple commands that transfer the data in a different format or presentation. For example, a container application could offer the following choices for creating links: Link Here, Link *Short Type Name* Here, and Create Shortcut Here.

For more information about using direct manipulation for moving, copying, and linking objects, see Chapter 5, "General Interaction Techniques."

The default appearance of a transferred object also depends on the destination container application. For most types of documents, display the data or content presentation of the object (or in the case of an OLE linked object, a representation of the content), rather than as an icon. If the user chooses Create Shortcut Here as the transfer operation, the transferred object is displayed as an icon. If the object cannot be displayed as content—for example, because it does not support OLE—the object is displayed as an icon.

Inserting New Objects

In addition to transferring objects, you can support user creation of OLE embedded or linked objects by generating a new object based on an existing object or object type and inserting the new object into the target container.

The Insert Object Command

Include an Insert Object command on the menu responsible for creating or importing new objects into a container, such as an Insert menu. If no such menu exists, use the Edit menu. When the user selects this command, display the Insert Object dialog box, as shown in Figure 11.5. This dialog box allows the user to generate new objects based on their object type or an existing file.

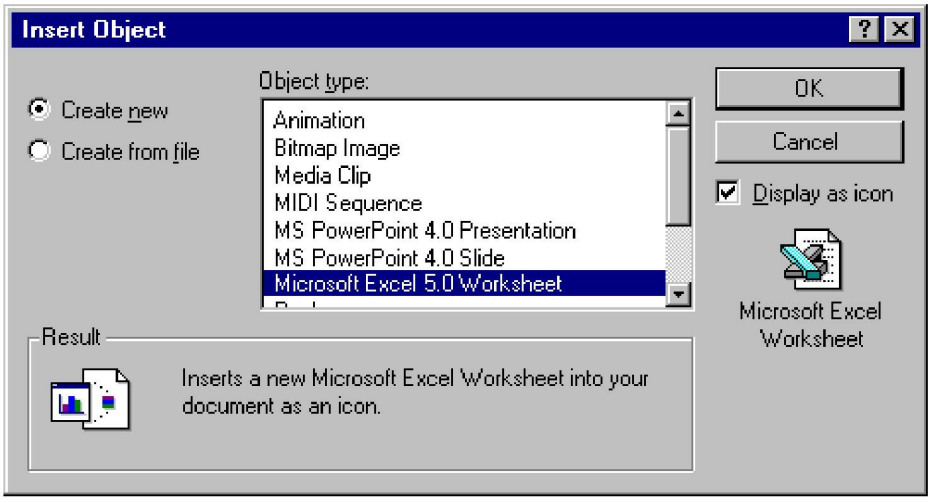

Figure 11.5 The Insert Object dialog box

The type list is composed of the type names of registered types. When the user selects a type from the list box and chooses the OK button, an object of the selected type is created and embedded.